PE-OPSD: INTERNALIZING PROMPT ENHANCEMENT INTO FLOW-MATCHING MODELS VIA ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-to-image users often provide concise and underspecified prompts, whereas generative models benefit from detailed textual conditions for reliable instruction following. Existing systems bridge this gap with Prompt Enhancers (PEs) that rewrite raw prompts at inference time, introducing additional latency and leaving prompt elaboration external to the generator. We instead view enhanced prompts as privileged training information and ask whether their benefits can be internalized. We propose Prompt-Enhanced On-Policy Self-Distillation (**PE-OPSD**) for text-to-image flow-matching models. During training, a raw-prompt student follows its own generation trajectory, while an enhanced-prompt teacher provides vector-field targets at the states visited by the student. This on-policy supervision distills the behavior induced by enhanced prompts into the raw-prompt student without requiring additional text–image pairs. At inference, both the PE and teacher are removed, and the student generates directly from raw prompts. Across multiple model families, PEs, and benchmarks, PE-OPSD achieves the strongest aggregate prompt fidelity among the evaluated baselines, yields positive aggregate visual appeal gains, and retains the base-model inference efficiency.

1 INTRODUCTION

User prompts are often concise descriptions rather than fully specified generation instructions (Xie et al., 2023; Hahn et al., 2024). A prompt such as "a girl reading under a tree" specifies the main subject and scene while leaving pose, lighting, composition, and visual style open. Although these unspecified choices admit multiple valid realizations, explicitly instantiating them can help generative models produce visually coherent outputs and follow the stated constraints more reliably. Despite substantial progress in visual fidelity and sampling efficiency (Zhang et al., 2023; Yin et al., 2024; Jiang et al., 2026b), reliable intent understanding remains difficult when prompts are short and underspecified (Huang et al., 2026). This creates a fundamental mismatch between concise user prompts and the detailed textual conditions preferred by generative models.

To bridge this gap, many industrial text-to-image systems employ a Prompt Enhancer (PE) before image generation (Team, 2025; Zhao et al., 2026a). Given a raw prompt, PE rewrites it into a longer description that makes implicit subject attributes, scene layouts and object relationships explicit. Prior works such as Promptist (Hao et al., 2023), BeautifulPrompt (Cao et al., 2023), and PromptEnhancer (Wang et al., 2025) have shown that prompt rewriting can improve human preference, text–image relevance, attribute binding, and compositional relationships (Ghosh et al., 2023). However, PE improves generation by modifying the input rather than improving the generator itself: prompt enhancement remains delegated to an additional inference-time module.

This deployment paradigm has practical and conceptual limitations. Practically, prompt rewriting introduces extra latency and computation, while longer prompts increase text-processing and conditioning overhead (Wang et al., 2023). Conceptually, an ideal generator should not require an external rewriter for every generation. It is therefore desirable for a generator to internalize the generation behavior induced by PEs and generate high-quality images directly from concise prompts.

This raises a natural question: ***Can a generator capture the generation benefits elicited by prompt enhancement while receiving only the raw prompt at inference time?*** Existing approaches exploit richer textual conditions mainly in two ways. Some replace original text conditions with expanded

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

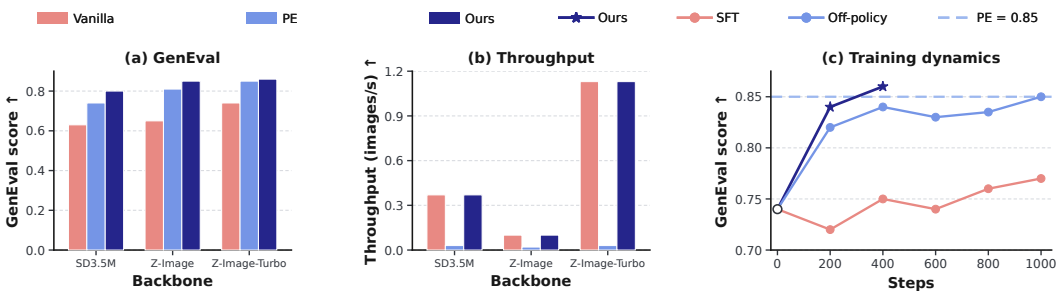

Figure 1: Comparison of generation quality, throughput, and training dynamics. PE-OPSD achieves higher GenEval scores than both Vanilla and PE while preserving the inference efficiency. Moreover, PE-OPSD achieves faster convergence and greater performance gains than SFT and off-policy distillation. Training dynamics is reported in `Z-Image-Turbo` training.

captions during training (Betker et al., 2023), which improves training data quality but does not directly teach the model to recover enhanced-prompt behavior from raw prompts. Others keep the generator fixed and continuously invoke PE at inference time (Cao et al., 2023; Mañas et al., 2024; Wang et al., 2025), which is effective but retains the deployment cost. Thus, less attention has been paid to internalizing the benefit of prompt enhancement into the generator itself.

In this work, we reinterpret prompt enhancement as a form of *privileged information*. Unlike conventional privileged information from additional modalities, an enhanced prompt elaborates the raw prompt while preserving its explicit semantic constraints. Crucially, we do not feed this privileged text to the student. Instead, it is used only to define a better-informed teacher condition, while the student must learn to reproduce the corresponding generation behavior from the raw prompt alone.

Building on this perspective, we propose Prompt-Enhanced On-Policy Self-Distillation (PE-OPSD) for text-to-image flow-matching models. Given a raw prompt p, the PE produces an enhanced prompt p^+. The student is conditioned only on p, while the teacher receives p^+ as privileged information. During training, the student follows its own vector field to generate on-policy flow trajectories. At the states visited by the student, the teacher provides target vector fields conditioned on p^+. By matching these targets under the raw-prompt condition, the student learns to approximate the generation behavior induced by enhanced prompts. Unlike supervised finetuning (SFT), PE-OPSD does not require constructing additional text-image pairs. Unlike off-policy distillation based on teacher trajectories, it supervises the student on its own raw-prompt trajectories, better aligning the training signal with the states encountered during raw-prompt generation. In this way, PE-OPSD converts PE from an inference-time module into a training-time supervision signal on flow dynamics. After training, both PE and teacher are removed, enabling direct generation from raw prompts without inference-time prompt rewriting.

Our contributions are summarized threefold:

- We reinterpret prompt enhancement as privileged information for text-to-image generation, studying how the benefits of enhanced prompts can be transferred to a model that observes only raw prompts at inference time.
- We propose Prompt-Enhanced On-Policy Self-Distillation (PE-OPSD) for text-to-image flow-matching models, where an enhanced-prompt teacher provides dense vector-field supervision on states visited by a raw-prompt student, thereby distilling enhanced-prompt generation behavior into a model conditioned only on raw prompts.
- We show that PE-OPSD outperforms inference-time PE, SFT, and off-policy distillation in aggregate prompt fidelity on all three main backbones, while retaining positive aggregate visual-appeal changes and base-model inference latency.

2 Related Work

Prompt enhancement for text-to-image generation. Prompt enhancement aims to bridge the gap between underspecified user prompts and the detailed textual conditions under which text-to-

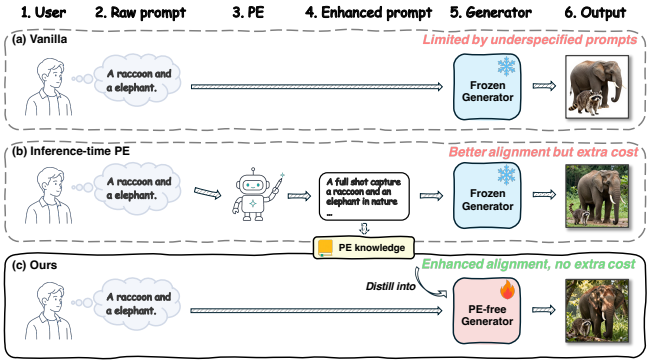

Figure 2: **Three different uses of prompt conditioning.** (a) Direct generation from raw prompts results in limited prompt alignment; (b) Inference-time PE improves prompt alignment by rewriting raw prompts, but introduces additional computational overhead; (c) Our PE-OPSD internalizes the PE knowledge into the generator, improving alignment without additional inference cost.

image models more reliably satisfy explicit prompt constraints and produce visually coherent outputs. Promptist (Hao et al., 2023) learns model-preferred prompts with reinforcement learning, BeautifulPrompt (Cao et al., 2023) trains a PE from low- and high-quality prompt pairs with visual feedback, and PromptEnhancer (Wang et al., 2025) further improves PE through chain-of-thought reasoning and fine-grained reward signals. However, these improvements are largely achieved outside the generator by training an external PE. In contrast, our work uses enhanced prompts only as training-time privileged information and distills their effect into the generator, enabling inference directly from raw prompts.

On-policy distillation. On-policy distillation (OPD) (Agarwal et al., 2024; Gu et al., 2024) mitigates train–inference mismatch by supervising the student on samples generated from its own current policy rather than from a fixed offline distribution. On-policy self-distillation (OPSD) (Zhao et al., 2026b) extends this idea by constructing asymmetric teacher and student views from the same base model, reducing the need for a separate stronger teacher. In recent visual and multimodal methods (Bi et al., 2026; Liu et al., 2026b; Yuan et al., 2026), this asymmetry is often induced by privileged information available only to the teacher during training, such as cropped regions, higher-resolution inputs, or visual reasoning traces. D-OPSD (Jiang et al., 2026a) further introduces this paradigm to step-distilled diffusion models, but relies on paired image–text data and a generator capable of accepting image-conditioned inputs. PE-OPSD differs from these methods by using enhanced prompts only as training-time privileged information, thereby transferring their effect to a raw-prompt flow-matching model without requiring external prompt enhancement at inference.

3 Methodology

3.1 Preliminaries

On-policy distillation. Let f_θ denote a student model and f_ϕ denote a teacher model. Conventional knowledge distillation (Hinton et al., 2015; Beyer et al., 2022) supervises the student on examples drawn from a fixed data distribution, which may differ from the states encountered by the student at inference. On-policy distillation instead evaluates the teacher on samples generated by the current student distribution. In a generic form, the objective can be written as

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{z\sim q_\theta(\cdot|c)}\left[D_{\mathrm{KL}}\big(f_\theta(z,c), f_\phi(z,c)\big)\right],$$

where c denotes the conditioning input, $q_\theta(\cdot \mid c)$ is the distribution induced by the current student.

On-policy self-distillation. On-policy self-distillation further removes the need for a separate teacher model by deriving teacher and student signals from asymmetric views of the same base model. Let c_s denote the student condition and c_t denote a stronger teacher condition, where c_t is the privileged information available only during training. The corresponding objective is

$$\mathcal{L}_{\mathrm{OPSD}}(\theta) = \mathbb{E}_{z\sim q_\theta(\cdot|c_s)}\left[D_{\mathrm{KL}}\big(f_\theta(z,c_s), f_{\bar\theta}(z,c_t)\big)\right],$$

where $f_{\bar\theta}$ denotes the teacher model, which may be a stop-gradient version of the base model or an exponential-moving-average (EMA) (Morales-Brotons et al., 2024) copy of the student model.

3.2 From Prompt Enhancement to Privileged Information

Prompt enhancement elicits conditional capability. Let p denote a raw user prompt and let E be a prompt enhancer that produces an enhanced prompt $p^+ = E(p)$. Figure 3 and Table 1 compare these conditions using the same frozen Z-Image-Turbo generator (Cai et al., 2025). Across the tested PEs, enhanced prompts improve the reported compositional scores while substantially increasing prompt length. By adding descriptive details to p while preserving its explicit constraints, PE provides a richer condition for generation.

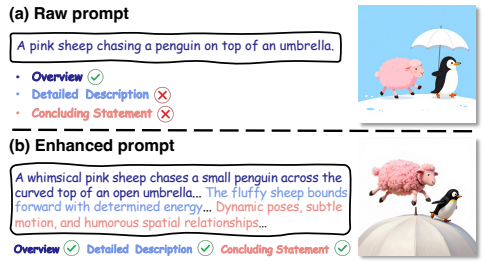

Figure 3: Comparison of raw and enhanced prompt, with the enhanced prompt providing denser and more detailed textual information.

Table 1: Comparison of the Base and inference-time PE on GenEval and GenEval2. †denotes `GPT-5.6 Sol` as PE. ‡denotes `PromptEnhancer` as PE.

Method	GenEval Task		GenEval2 Task		
	Token Len.	GE	Token Len.	$GE2_{AM}$	$GE2_{GM}$
Base	11.02	0.737	7.92	0.783	0.341
PE-GPT†	152.61	0.850	135.11	0.843	0.479
Δ (vs Base)	13.9×	+0.113	17.1×	+0.060	+0.138
PE-7B‡	198.29	0.746	170.66	0.802	0.383
Δ (vs Base)	18.0×	+0.009	21.5×	+0.019	+0.042
PE-32B‡	155.82	0.859	148.66	0.849	0.492
Δ (vs Base)	14.1×	+0.122	18.8×	+0.066	+0.151

We view p^+ as a PE-selected elaboration of p, intended to preserve the explicit semantics of p while adding one plausible realization of otherwise unspecified attributes, composition, and scene details. Because prompt enhancement changes only the conditioning input while keeping the generator fixed (Hao et al., 2023; Wang et al., 2025), the performance gap between p and p^+ shows that the pretrained generator can better realize the requested content when conditioned on a more detailed description. This does not imply that the generator can recover the missing details from p alone. Instead, it shows that the improved generation behavior is attainable under enhanced prompts. We therefore seek to internalize this enhanced-prompt behavior into a model conditioned only on p.

Enhanced prompts as native textual privileged information. We cast this objective through the lens of learning with privileged information. During training, p^+ provides the teacher with a richer PE-generated condition associated with the same raw prompt, while the student remains conditioned only on p; at deployment, p^+ is unavailable. Importantly, the teacher's advantage arises from asymmetric conditioning rather than greater model capacity. Prompt enhancement therefore constitutes a native form of textual privileged information: both p and p^+ are processed by the same architecture through its existing text-conditioning pathway, without introducing an auxiliary modality.

This differs from image-privileged formulations such as D-OPSD (Jiang et al., 2026a), which construct the teacher condition by jointly encoding the prompt and a paired target image with a multimodal encoder. PE-OPSD instead derives the privileged condition from prompt enhancement alone, requiring neither paired target images nor modifications to the model's conditioning interface. Crucially, our goal is not to reconstruct p^+ or imitate the PE itself, but to distill the generation dynamics elicited by p^+ into a student that observes only p. We next formalize this principle for flow-matching generators.

3.3 PE-OPSD: Prompt-Enhanced On-Policy Self-Distillation

Overview. Figure 4 illustrates the training and inference pipelines of PE-OPSD. Let $\mathcal{D} = \{(p, p^+)\}$ denote the training prompt pairs set, where $p^+ = E(p)$ is obtained through PE. The student and teacher share the same architecture and are initialized from the same pretrained parameters, $\theta = \bar{\theta} = \theta_{\text{pre}}$. During training, the student generates trajectories conditioned only on p, while the teacher provides enhanced-prompt supervision on the states visited by the student. After training, the behavior induced by enhanced prompts is internalized into the generator, enabling direct raw-prompt generation without invoking PE at inference time.

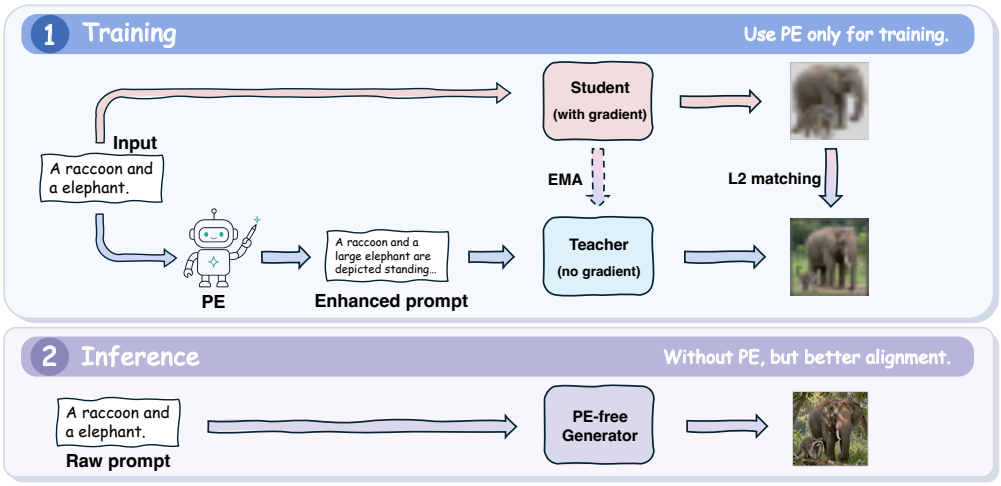

Figure 4: **Overview of PE-OPSD.** At training, the student generates rollouts from raw prompts, while an EMA teacher provides enhanced-prompt supervision on the same rollout states. At inference, the trained student generates directly from raw prompts without inference-time PE.

On-policy sampling under raw prompts. We collect training states from the current student's own generation process. Let $1 = t_K > t_{K-1} > \cdots > t_0 = 0$ denote the discrete denoising schedule, and let $\Delta t_k = t_k - t_{k-1} > 0$. Starting from $x_{t_K} \sim \mathcal{N}(0, I)$, the student follows the Euler updates

$$x_{t_{k-1}} = x_{t_k} - \Delta t_k v_\theta(x_{t_k}, t_k, p), \qquad k = K, \ldots, 1. \tag{1}$$

We supervise the student at the visited states $\tau = \{x_{t_k}\}_{k=1}^K$. Because these states are generated by the current student under the raw condition p, τ follows the state induced by the current student under the training sampler. In contrast, a trajectory generated by the teacher under p^+ would follow a different state distribution and thus provide off-policy supervision for the raw-prompt student.

Distillation from enhanced-prompt supervision. At each student-visited state x_{t_k}, the student and teacher are evaluated at the same state and time but under different prompts:

$$v_k^S = v_\theta(x_{t_k}, t_k, p), \qquad v_k^T = v_{\bar{\theta}}(x_{t_k}, t_k, p^+). \tag{2}$$

Because PE-OPSD operates with deterministic flow trajectories, we directly regress the teacher's vector field rather than introducing a stochastic transition kernel. We consider matching velocities v, one-step transitions μ, or predicted clean latents $\hat{x}_0$. Under the linear flow interpolation $x_t = (1-t)x_0 + t\epsilon$ (Lipman et al., 2022), with $\epsilon \sim \mathcal{N}(0, I)$, the latter two targets are

$$\mu_k^b = x_{t_k} - \Delta t_k \, v_k^b, \qquad \hat{x}_{0,k}^b = x_{t_k} - t_k \, v_k^b, \qquad b \in \{S, T\}. \tag{3}$$

Since both μ_k and $\hat{x}_{0,k}$ share the same state x_{t_k}, their squared-error objectives reduce to a weighted velocity mismatch. We therefore express all three variants using a unified objective:

$$\mathcal{L}_{\text{PE-OPSD}}(\theta; \bar{\theta}) = \mathbb{E}_{\substack{(p,p^+)\sim\mathcal{D} \\ z\sim\mathcal{N}(0,I)}} \left[\frac{1}{K} \sum_{k=1}^{K} \omega(t_k) \left\| v_k^S - \text{sg}\left[v_k^T\right] \right\|_2^2 \right], \tag{4}$$

where $\omega_v(t_k) = 1$, $\omega_{x_0}(t_k) = t_k^2$, and $\omega_\mu(t_k) = (t_{k-1} - t_k)^2 = (\Delta t_k)^2$, respectively. $\text{sg}[\cdot]$ denotes stop-gradient. We refer to these variants as $v-$loss, x_0-loss, and $\mu-$loss. The three variants share the same pointwise optimum in velocity space but differ in timestep weighting. Appendix A provides a trajectory-level KL interpretation and its connection to our deterministic matching objectives.

During optimization, the teacher predictions are detached, and gradients pass only through the student predictions in Equation 2. After updating the student from θ_n to θ_{n+1}, we update the teacher through EMA as

$$\bar{\theta}_{n+1} \leftarrow \gamma \, \bar{\theta}_n + (1-\gamma) \, \theta_{n+1}, \qquad 0 \le \gamma < 1, \tag{5}$$

where γ is the EMA decay rate. This provides a temporally smoothed teacher that evolves with the student while retaining enhanced-prompt conditioning.

Training recipe. Algorithm 1 summarizes the training procedure. We first construct prompt pairs set $\mathcal{D} = \{(p, p^+)\}$ by applying the off-the-shelf PE offline, avoiding PE calls during training. Each iteration rollouts the student using the raw prompt p, and evaluates the enhanced-prompt teacher at the visited states. The student is optimized to match the stop-gradient teacher velocities, after which the teacher is updated by EMA. After training, only the student is retained and generation proceeds directly from raw prompts without invoking the PE or teacher.

Algorithm 1 PE-OPSD Training

Require: Prompt pairs set $\mathcal{D} = \{(p, p^+)\}$; pretrained model θ_{pre}; schedule $\{t_0, \cdots, t_K\}$; EMA decay rate γ

$\theta \leftarrow \theta_{\text{pre}}$, $\bar{\theta} \leftarrow \theta_{\text{pre}}$
for each training iteration **do**
  Sample a mini-batch $(p, p^+) \sim \mathcal{D}$ and $x_{t_K} \sim \mathcal{N}(0, I)$
  Initialize $\mathcal{L} \leftarrow 0$
  for each $k = K, \ldots, 1$ **do**
    $\mathcal{L} \leftarrow \mathcal{L} + \frac{\omega(t_k)}{K} \| v_\theta(x_{t_k}, t_k, p) - \text{sg}[v_{\bar{\theta}}(x_{t_k}, t_k, p^+)] \|_2^2$ ▷ $\mu-$loss by default
    if $k > 1$ **then**
      $x_{t_{k-1}} \leftarrow \text{sg}[x_{t_k} - \Delta t_k v_\theta(x_{t_k}, t_k, p)]$ ▷ Rollout for one step
    end if
  end for
  $\theta \leftarrow \text{OptimizerStep}(\theta, \nabla_\theta \mathcal{L})$
  $\bar{\theta} \leftarrow \gamma \bar{\theta} + (1 - \gamma)\theta$ ▷ EMA update
end for
return θ

4 Experiments

4.1 Experimental Setups

Models and baselines. Our main experiments use SD3.5-M (Esser et al., 2024), Z-Image, and Z-Image-Turbo, comparing PE-OPSD against the unmodified model (Base), inference-time prompt enhancement (Base+PE), supervised fine-tuning (SFT), and off-policy distillation. For SFT, we first generate teacher images conditioned on enhanced prompts offline and then fine-tune the student on the images. Off-policy distillation uses the same teacher and distillation objective as PE-OPSD but collects training states from enhanced-prompt teacher trajectories rather than raw-prompt student trajectories. For fairness, all trainable methods use the same initialization, training data, optimization configuration, and number of training steps. We further evaluate FLUX.2-klein-base, FLUX.2-klein (Black Forest Labs, 2026), and QwenImage-2512 (Zhao et al., 2026a) to assess scalability. We use `GPT-5.6 Sol` as the default PE and evaluate PE generalization on Z-Image with `PromptEnhancer-7B/32B` (Wang et al., 2025).

Benchmarks and metrics. We evaluate compositional generation on GenEval (GE, Ghosh et al. (2023)) and GenEval2 (GE2, Kamath et al. (2025)), reporting GE and $\text{GE2}_{\text{GM/AM}}$, alongside CLIP score (Hessel et al., 2021), PickScore (Kirstain et al., 2023), and aesthetics (Schuhmann, 2022) on both benchmarks. We summarize improvement over Base as

$$\mathbb{I}_c = \frac{100\%}{|\mathcal{M}_c|} \sum_{m \in \mathcal{M}_c} \left(\frac{\text{Score}_{\text{method}} - \text{Score}_{\text{Base}}}{\text{Score}_{\text{Base}}} \right), \quad c = \{\text{PF}, \ \text{VA}\} \tag{6}$$

where $\mathcal{M}_{\text{PF}}$ contains GE, GE2_{GM}, GE2_{AM}, and CLIP scores on both benchmarks, while $\mathcal{M}_{\text{VA}}$ contains PickScore and aesthetics on both benchmarks. These indices measure **P**rompt **F**idelity and **V**isual **A**ppeal improvements, respectively. Out-of-domain evaluation uses DPG-bench (Hu et al., 2024), T2I-CompBench++ (Huang et al., 2025), and EvalMuse (Han et al., 2026).

Configuration. For GenEval, we adopt the training and evaluation splits released by Flow-GRPO (Liu et al., 2026c). For GenEval2, we train on the official 20K synthetic prompts and evaluate on the 800 officially released prompts. All models are trained on the same mixture of the GenEval and GenEval2 training sets. To ensure a fair comparison, we use a unified training recipe across all models and methods. Full configuration details and experimental setups are provided in Appendix B.

Table 2: **Main results across different models.** We use `GPT-5.6 Sol` as PE. PickScore is normalized by 26; Aes. denotes aesthetics; **Bold:** best; Underlined: second-best.

Method	GenEval (GE) Task				GenEval2 (GE2) Task					$\mathbb{I}_{\mathrm{PF}}$	$\mathbb{I}_{\mathrm{VA}}$
	GE	PickScore	CLIP	Aes.	$\mathrm{GE2}_{\mathrm{GM}}$	$\mathrm{GE2}_{\mathrm{AM}}$	PickScore	CLIP	Aes.		
SD3.5-L	0.604	0.862	0.288	5.286	0.213	0.660	0.869	0.315	5.521	-	-
FLUX.1-dev	0.618	0.899	0.280	5.624	0.179	0.643	0.895	0.298	5.909	-	-
SD3.5-M (2.5B)											
Base	0.628	0.878	0.289	5.319	0.176	0.633	0.882	0.313	5.579	0.00%	0.00%
Base+PE	0.743	0.892	0.291	5.439	0.220	0.680	0.887	0.312	5.678	10.22%	1.55%
SFT	0.718	0.889	0.295	**5.462**	0.207	0.656	0.886	0.310	5.702	7.34%	1.65%
Off-Policy Distillation	0.770	0.892	**0.296**	5.407	0.223	0.671	0.890	0.316	5.567	11.74%	0.99%
PE-OPSD (Ours)	**0.797**	**0.896**	0.295	5.401	**0.226**	**0.682**	**0.893**	**0.318**	**5.708**	**13.35%**	**1.79%**
Δ (vs Base)	+0.169	+0.018	+0.006	+0.082	+0.050	+0.049	+0.011	+0.005	+0.129	+13.35%	+1.79%
Z-Image (6B)											
Base	0.650	0.875	0.288	5.288	0.306	0.761	0.871	0.319	5.421	0.00%	0.00%
Base+PE	0.810	0.899	0.299	**5.433**	0.404	0.827	0.893	0.326	5.638	14.27%	3.00%
SFT	0.821	0.893	**0.302**	5.426	0.403	**0.834**	0.887	0.326	5.466	14.93%	1.83%
Off-Policy Distillation	0.824	0.901	0.300	5.432	0.395	0.828	0.897	0.327	5.629	14.27%	3.12%
PE-OPSD (Ours)	**0.846**	**0.902**	**0.302**	5.423	**0.406**	0.831	**0.900**	**0.330**	**5.650**	**16.07%**	**3.30%**
Δ (vs Base)	+0.196	+0.027	+0.014	+0.135	+0.100	+0.070	+0.029	+0.011	+0.229	+16.07%	+3.30%
Z-Image-Turbo (6B)											
Base	0.737	0.908	0.291	5.289	0.341	0.783	0.901	0.320	5.436	0.00%	0.00%
Base+PE	0.850	0.918	**0.298**	**5.306**	0.479	0.843	0.911	0.325	**5.561**	13.49%	**1.21%**
SFT	0.766	0.893	0.295	5.218	0.413	0.788	0.884	0.317	5.398	5.22%	-1.40%
Off-Policy Distillation	0.851	0.916	0.296	5.251	0.437	0.821	0.910	0.325	**5.561**	10.35%	0.87%
PE-OPSD (Ours)	**0.863**	**0.919**	**0.298**	5.274	**0.501**	**0.844**	**0.913**	**0.326**	5.547	**15.22%**	1.08%
Δ (vs Base)	+0.126	+0.011	+0.007	-0.015	+0.160	+0.061	+0.012	+0.006	+0.111	+15.22%	+1.08%



Figure 5: **Qualitative comparison and human preference study on Z-Image-Turbo.** We use `GPT-5.6 Sol` as PE. **Left:** visual comparisons among Base, PE, SFT, Off-policy distillation, and our method. **Right:** pairwise human preferences against Base in prompt fidelity and visual appeal.

4.2 Main Experiments

Main results. Table 2 shows that PE-OPSD consistently achieves the best prompt fidelity across all three backbones. It obtains the highest GE, $\mathrm{GE2}_{\mathrm{GM}}$, and $\mathbb{I}_{\mathrm{PF}}$, outperforming SFT, off-policy distillation, and even Base+PE, which retains the PE at inference. Relative to Base, PE-OPSD improves aggregate prompt fidelity by 13.35%, 16.07%, and 15.22% on SD3.5-M, Z-Image, and Z-Image-Turbo, respectively. Meanwhile, $\mathbb{I}_{\mathrm{VA}}$ remains positive on every backbone, indicating that PE-OPSD improves prompt fidelity without degrading aggregate visual appeal.

Applicability to post-trained models. PE-OPSD remains effective on the already post-trained Z-Image-Turbo (post-trained by Decoupled DMD (Liu et al., 2026a) and DMDR (Jiang et al., 2026b)). SFT yields smaller fidelity gains and a 1.40% decline in visual appeal, whereas PE-OPSD improves both. These results support its applicability to models that have already undergone post-training.

Qualitative results and human preference study. Figure 5 presents visual examples and human preference study results. In pairwise comparisons against Base, human raters prefer PE-OPSD in 66% of comparisons for prompt fidelity and 58% for visual appeal. These are the highest observed preference rates among the evaluated methods, complementing the quantitative results. Human study details are provided in Appendix C.

Efficiency. Table 3 compares prompt fidelity and latency on Z-Image. Under both evaluated PEs, PE-OPSD improves all three fidelity metrics over inference-time PE while retaining the measured latency of Base. It is 1.96× and 4.51× faster than deploying `PromptEnhancer-7B` and `PromptEnhancer-32B`, respectively. Thus, PE-OPSD transfers the benefits of prompt enhancement to the generator without requiring a PE for each inference request. More results about training costs are reported in Appendix D.2.

Table 3: **Performance and latency.**

Method	GE	$GE2_{GM}$	$GE2_{AM}$	Lat. (s)
`PromptEnhancer-7B as PE`				
Base	0.650	0.306	0.761	9.69
Base+PE	0.684	0.375	0.801	19.00
Base+Ours	0.782	0.422	0.819	9.69
Δ (vs +PE)	+0.098	+0.047	+0.018	1.96×
`PromptEnhancer-32B as PE`				
Base	0.650	0.306	0.761	9.69
Base+PE	0.815	0.470	0.844	43.74
Base+Ours	0.868	0.476	0.846	9.69
Δ (vs +PE)	+0.053	+0.006	+0.002	4.51×

Table 4: **Result across different PEs. Bold:** best; Underlined: second-best.

Method	GenEval (GE) Task				GenEval2 (GE2) Task					$\mathbb{I}_{PF}$	$\mathbb{I}_{VA}$
	GE	PickScore	CLIP	Aes.	$GE2_{GM}$	$GE2_{AM}$	PickScore	CLIP	Aes.		
`GPT-5.6 Sol as PE`											
Z-Image	0.650	0.875	0.288	5.288	0.306	0.761	0.871	0.319	5.421	0.00%	0.00%
Z-Image+PE	0.810	0.899	0.299	**5.433**	0.404	0.827	0.893	0.326	5.638	14.27%	3.00%
SFT	0.821	0.893	**0.302**	5.426	0.403	**0.834**	0.887	0.326	5.466	14.93%	1.83%
Off-Policy Distillation	0.824	0.901	0.300	5.432	0.395	0.828	0.897	0.327	5.629	14.27%	3.12%
PE-OPSD (Ours)	**0.846**	**0.902**	**0.302**	5.423	**0.406**	0.831	**0.900**	**0.330**	**5.650**	**16.07%**	**3.30%**
Δ (vs Base)	+0.196	+0.027	+0.014	+0.135	+0.100	+0.070	+0.029	+0.011	+0.229	+16.07%	+3.30%
`PromptEnhancer-7B as PE`											
Z-Image	0.650	0.875	0.288	5.288	0.306	0.761	0.871	0.319	5.421	0.00%	0.00%
Z-Image+PE	0.684	0.884	0.287	**5.464**	0.375	0.801	0.879	0.318	5.589	6.28%	2.09%
SFT	0.763	0.887	**0.299**	5.424	0.400	**0.819**	0.881	0.325	5.424	12.29%	1.29%
Off-Policy Distillation	0.767	0.896	0.296	5.410	0.380	0.804	0.889	0.324	5.573	10.44%	2.39%
PE-OPSD (Ours)	**0.782**	**0.898**	0.296	5.417	**0.422**	**0.819**	**0.892**	**0.327**	**5.606**	**14.22%**	**2.72%**
Δ (vs Base)	+0.132	+0.023	+0.008	+0.129	+0.116	+0.058	+0.021	+0.008	+0.185	+14.22%	+2.72%
`PromptEnhancer-32B as PE`											
Z-Image	0.650	0.875	0.288	5.288	0.306	0.761	0.871	0.319	5.421	0.00%	0.00%
Z-Image+PE	0.815	0.889	0.294	**5.381**	0.470	0.844	0.879	0.325	**5.515**	18.77%	1.50%
SFT	0.829	0.889	**0.302**	**5.381**	0.440	0.845	0.877	0.329	5.373	18.07%	0.79%
Off-Policy Distillation	0.842	0.895	0.300	5.363	0.459	0.843	0.884	0.331	5.455	19.65%	1.46%
PE-OPSD (Ours)	**0.868**	**0.897**	0.300	5.352	**0.476**	**0.846**	**0.888**	**0.334**	5.496	**21.83%**	**1.76%**
Δ (vs Base)	+0.218	+0.022	+0.012	+0.064	+0.170	+0.085	+0.017	+0.015	+0.075	+21.83%	+1.76%

Table 5: **Results on out-of-domain benchmarks.** For DPG-bench, we report Global (Glo.), Entity (Ent.), Attribute (Attr.), Relation (Rela.), Other and Overall⋆. For T2I-CompBench++, we report Color, Shape, Textual (Tex.), Numeracy (Num.), Complex (Comp.), Spatial (Spa.), 3D Spatial (3D Spa.), Non-spatial (Non-spa.) and Overall⋆. **Bold:** best in Overall⋆.

Method	DPG-bench						T2I-CompBench++									EvalMuse
	Overall⋆	Glo.	Ent.	Attr.	Rela.	Other	Overall⋆	Color	Shape	Tex.	Num.	Comp.	Spa.	3D Spa.	Non-spa.	Overall⋆
SD3.5-M	**83.9**	84.6	89.6	88.1	93.0	80.9	0.51	0.80	0.54	0.74	0.59	0.37	0.32	0.36	0.31	3.199
+Ours	83.8	83.9	89.3	88.4	93.0	82.1	**0.55**	0.83	0.60	0.73	0.63	0.39	0.46	0.42	0.31	**3.368**
Z-Image	86.3	83.5	91.7	90.1	94.6	87.8	0.53	0.85	0.59	0.79	0.63	0.40	0.31	0.37	0.31	3.340
+Ours	**87.0**	82.6	92.0	90.2	94.5	89.4	**0.58**	0.87	0.61	0.81	0.71	0.42	0.36	0.41	0.32	**3.561**
Z-Image-Turbo	84.5	78.6	91.2	88.3	93.2	88.4	0.53	0.81	0.56	0.75	0.69	0.40	0.36	0.41	0.31	3.515
+Ours	**85.0**	77.8	91.2	88.9	93.9	88.1	**0.56**	0.89	0.53	0.76	0.72	0.40	0.52	0.42	0.31	**3.534**

4.3 Generalization

Different PEs. On Z-Image, PE-OPSD achieves the highest $\mathbb{I}_{PF}$ and $\mathbb{I}_{VA}$ with each of the three tested PEs (Table 4), demonstrating that its benefits extend beyond the default PE. The largest aggregate fidelity gain is obtained with `PromptEnhancer-32B`, while `GPT-5.6 Sol` yields the largest visual appeal gain, suggesting that PE choice affects the balance between these objectives.

Out-of-domain generalization. Table 5 shows that all three backbones improve their Overall scores on T2I-CompBench++ and EvalMuse, with consistent gains in spatial relations on T2I-CompBench++. On DPG-Bench, which features long and detailed prompts, PE-OPSD largely preserves the base models' performance. Overall, PE-OPSD generalizes across OOD benchmarks, improving compositional fidelity while preserving performance on already detailed prompts.

Scalability. We further evaluate PE-OPSD on the larger 9B FLUX.2-klein variants (Black Forest Labs, 2026) and the 20B QwenImage-2512 (Zhao et al., 2026a), as summarized in Figure 6 (complete results are in Appendix D.4). Across three models, PE-OPSD improves upon the base model and inference-time PE on aggregate prompt fidelity. It also remains competitive across aggregate visual appeal over Base. These results demonstrate that PE-OPSD scales effectively to larger models.

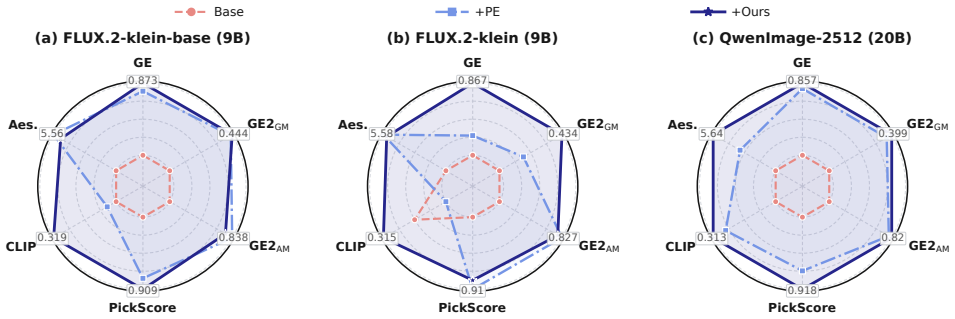

Figure 6: **Scaling to larger models.** PickScore, CLIP, and Aesthetics are averaged over GenEval and GenEval2. For visualization, each metric is independently normalized to $[0.3, 1.0]$.

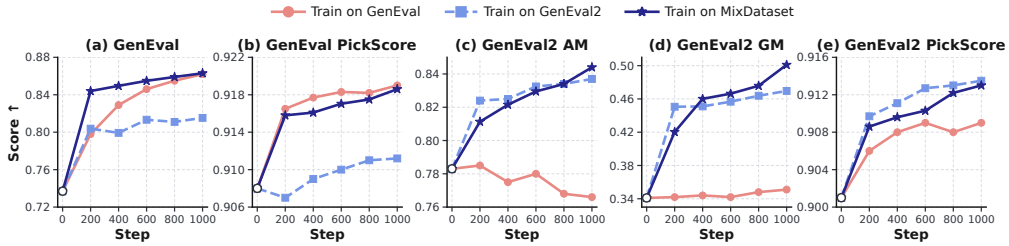

Figure 7: **Ablation study results on different training datasets.** We use MixDataset by default.

Table 6: Ablation study results on different loss functions, which differ only in their timestep weighting.

Loss	GE	$GE2_{GM}$	$GE2_{AM}$
x_0	0.863	0.474	0.834
v	0.866	0.487	0.838
μ	0.863	0.501	0.844

Table 7: Ablation study results on CFG settings. †denotes the unconditional branch is detached. Time (s): training times for one step.

Settings	GE	$GE2_{GM}$	$GE2_{AM}$	Time
CFG=4.0	0.834	0.430	0.835	99.13
CFG=4.0†	0.835	0.421	0.834	73.39
w/o CFG	0.846	0.406	0.831	49.15

Table 8: Ablation study results on the steps for training. Time (s): training times for one step.

Step	GE	$GE2_{GM}$	$GE2_{AM}$	Time
2	0.862	0.469	0.835	7.47
4	0.863	0.501	0.844	14.31
8	0.863	0.483	0.837	28.16

4.4 Ablation Studies

Loss. Among losses differing only in timestep weighting, μ-loss achieves the highest GE2 scores while remaining close to v-loss on GE (Table 6), motivating its use as the default.

CFG. Omitting CFG roughly halves training time and improves GE, but lowers GE2 scores (Table 7). Detaching the unconditional branch partially reduces the computational overhead. We omit CFG by default for training efficiency.

On-policy training step. We conduct this ablation study in `Z-Image-Turbo`. As shown in Table 8, 4 rollout steps achieve the highest GE2 scores at approximately half the iteration time of 8 steps, with nearly identical GE across the tested settings. We therefore use four steps by default.

Training data. As shown in Figure 7, training on a single dataset leads to dataset-specific specialization. GenEval-only training performs well on GenEval but transfers poorly to GenEval2, whereas GenEval2-only training compromises performance on GenEval. In contrast, MixDataset achieves the best balance across both benchmarks and is therefore used as our default training set.

5 Conclusion

We introduced PE-OPSD, an on-policy self-distillation framework that treats enhanced prompts as privileged training information. PE-OPSD transfers the generation behavior induced by enhanced prompts to a raw-prompt generator through supervision on student-generated trajectories. Across three main backbones, it improves aggregate prompt fidelity by 13.35%-16.07% over Base and exceeds inference-time PE, SFT, and off-policy distillation while retaining base-model latency. Results with alternative PEs, larger models, and out-of-domain benchmarks further support its applicability.

AI USE STATEMENT

The authors determined the research questions, methodology, experimental design, analysis, and claims of this work. Generative AI tools were used to assist with manuscript preparation, including grammar checking, sentence-level editing, and polishing the writing for clarity and readability. We also used LLM (GPT-5.6 Sol) as PE to generate synthetic datasets used in our experiments. All AI-assisted outputs, including revised text, prompts, and generated synthetic data, were reviewed, filtered, and validated by the authors. Generative AI tools were not used to autonomously produce scientific claims, conduct final analyses, or replace author judgment. The authors take full responsibility for the final content of the paper, including the text, experiments, results, and conclusions.

ETHICS STATEMENT

This work uses synthetic datasets and includes a human preference study in which participants compare pairs of images. The study is designed to collect perceptual judgments rather than sensitive personal information, and the responses are analyzed only in aggregate. All generated data and human annotations are reviewed and used only for research evaluation.

REPRODUCIBILITY STATEMENT

We provide details of the experimental setup, model configurations, evaluation metrics, and implementation procedures in the paper and appendix. For synthetic data generation, we will release the prompts and the generated datasets where permitted. For the human preference study, we report the details in the appendix. We will also release code and configuration files to facilitate reproduction of the main experiments and analyses.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10915–10924. IEEE, 2022.

Jinhe Bi, Peng Liao, Zengjie Jin, Volker Tresp, Fei Shen, Yunpu Ma, Tat-Seng Chua, et al. Opd-v: Visual on-policy self-distillation with modality balance. *arXiv preprint arXiv:2608.05131*, 2026.

Black Forest Labs. FLUX.2 [klein]: Towards Interactive Visual Intelligence. https://bfl.ai/blog/flux2-klein-towards-interactive-visual-intelligence, 2026.

Huanqia Cai, Sihan Cao, Ruoyi Du, Peng Gao, Aiming Hao, Steven Hoi, Zhaohui Hou, Shijie Huang, Dengyang Jiang, Yuming Jiang, et al. Z-image: An efficient image generation foundation model with single-stream diffusion transformer. *arXiv preprint arXiv:2511.22699*, 2025.

Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. Beautifulprompt: Towards automatic prompt engineering for text-to-image synthesis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1–11, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Robert M Gray. *Entropy and information theory*. Springer Science & Business Media, 2011.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Meera Hahn, Wenjun Zeng, Nithish Kannen, Rich Galt, Kartikeya Badola, Been Kim, and Zi Wang. Proactive agents for multi-turn text-to-image generation under uncertainty. *arXiv preprint arXiv:2412.06771*, 2024.

Shuhao Han, Haotian Fan, Jiachen Fu, Liang Li, Tao Li, Junhui Cui, Yunqiu Wang, Yang Tai, Jingwei Sun, Chun-Le Guo, et al. Evalmuse-40k: A fine-grained benchmark with comprehensive human annotations for text-to-image generation model alignment evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4583–4591, 2026.

Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:66923–66939, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Kaiyi Huang, Chengqi Duan, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench++: An enhanced and comprehensive benchmark for compositional text-to-image generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3563–3579, 2025.

Zijian Huang, Jay Zhangjie Wu, Zian Wang, Tianshi Cao, Jiasi Chen, Sanja Fidler, Huan Ling, and Xuanchi Ren. Ape: Agentic prompt enhancer for image generation and editing. *arXiv preprint arXiv:2606.00204*, 2026.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, et al. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026a.

Dengyang Jiang, Dongyang Liu, Zanyi Wang, Qilong Wu, Liuzhuozheng Li, Hengzhuang Li, Xin Jin, Changsheng Lu, Zhen Li, Mengmeng Wang, et al. Distribution matching distillation meets reinforcement learning. In *European Conference on Computer Vision*, pp. 281–299. Springer, 2026b.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. Geneval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026.

Mingfeng Lin, Chengfei Cai, Lin Xu, Yuxiang Wei, and Liang Han. Dreopd: Degraded-reference extrapolative on-policy distillation for flow-matching models. *arXiv preprint arXiv:2608.09233*, 2026.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Dongyang Liu, Gao Peng, David Liu, DU DU, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Steven HOI, et al. Decoupled dmd: Cfg augmentation as the spear, distribution matching as the shield. In *International Conference on Learning Representations*, volume 2026, pp. 140643–140666, 2026a.

Hongyu Liu, Chun Wang, Feng Gao, Xuanhua He, Yue Ma, Ziyu Wan, Yong Zhang, Xiaoming Wei, and Qifeng Chen. Opsd-v: On-policy self-distillation for post-training few-step autoregressive video generators. *arXiv preprint arXiv:2607.08766*, 2026b.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026c.

Oscar Mañas, Pietro Astolfi, Melissa Hall, Candace Ross, Jack Urbanek, Adina Williams, Aishwarya Agrawal, Adriana Romero-Soriano, and Michal Drozdzal. Improving text-to-image consistency via automatic prompt optimization. *arXiv preprint arXiv:2403.17804*, 2024.

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

Bowen Ping, Xiangxin Zhou, Penghui Qi, Minnan Luo, Liefeng Bo, and Tianyu Pang. Flow-dppo: Divergence proximal policy optimization for flow matching models. *arXiv preprint arXiv:2606.11025*, 2026.

Christoph Schuhmann. Laion-aesthetics, 2022. URL https://laion.ai/blog/laion-aesthetics/.

Tencent Hunyuan Foundation Model Team. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Linqing Wang, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Jiale Tao, Qixun Wang, Ruihuang Li, Comi Chen, et al. Promptenhancer: A simple approach to enhance text-to-image models via chain-of-thought prompt rewriting. *arXiv preprint arXiv:2509.04545*, 2025.

Yunlong Wang, Shuyuan Shen, and Brian Y Lim. Reprompt: Automatic prompt editing to refine ai-generative art towards precise expressions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–29, 2023.

Yutong Xie, Zhaoying Pan, Jinge Ma, Luo Jie, and Qiaozhu Mei. A prompt log analysis of text-to-image generation systems. In *Proceedings of the ACM Web Conference 2023*, pp. 3892–3902, 2023.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824. IEEE, 2023.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026a.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026b.

A From Trajectory KL to Deterministic Matching

The main text directly presents the deterministic objective used in training. Here, we provide a probabilistic interpretation through trajectory-level distillation and show how it motivates a deterministic matching surrogate on student-visited states.

Trajectory-level on-policy distillation. For a prompt pair (p, p^+), let the student and teacher trajectory distributions be

$$\Pi_\theta^S(\tau \mid p) = p(x_{t_K}) \prod_{k=1}^{K} \pi_{\theta,k}^S(x_{t_{k-1}} \mid x_{t_k}, p), \tag{7}$$

$$\Pi_{\bar{\theta}}^T(\tau \mid p^+) = p(x_{t_K}) \prod_{k=1}^{K} \pi_{\bar{\theta},k}^T(x_{t_{k-1}} \mid x_{t_k}, p^+), \tag{8}$$

where both distributions share the same initial noise prior. A trajectory-level distillation objective is

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{(p,p^+)\sim\mathcal{D}} \left[D_{\text{KL}}\big(\Pi_\theta^S(\cdot \mid p) \,\big\|\, \Pi_{\bar{\theta}}^T(\cdot \mid p^+)\big)\right]. \tag{9}$$

By the chain rule for KL divergence, this objective decomposes as

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{(p,p^+)\sim\mathcal{D}} \left[\sum_{k=1}^{K} \mathbb{E}_{x_{t_k}\sim d_{\theta,k}(\cdot|p)} \left[D_{\text{KL}}\left(\pi_{\theta,k}^S \,\middle\|\, \pi_{\bar{\theta},k}^T\right)\right]\right], \tag{10}$$

where $d_{\theta,k}(\cdot \mid p)$ is the state distribution induced by the current raw-prompt student. Thus, the teacher is queried at states visited by the student rather than states from its own enhanced-prompt trajectory.

Conditional Gaussian transitions. At a fixed student-visited state (x_{t_k}, t_k), consider Gaussian transitions with shared covariance (Liu et al., 2026c; Li et al., 2026; Lin et al., 2026):

$$\pi_k^b = \mathcal{N}(\mu_k^b, \Sigma_k), \qquad \mu_k^b = x_{t_k} - \Delta t_k v_k^b, \qquad b \in \{S, T\}, \tag{11}$$

where $\Sigma_k \succ 0$ is shared by the student and teacher. Their conditional KL divergence is

$$D_{\text{KL}}\big(\pi_k^S \,\|\, \pi_k^T\big) = \frac{1}{2} \left\|\mu_k^S - \mu_k^T\right\|_{\Sigma_k^{-1}}^2 = \frac{(\Delta t_k)^2}{2} \left\|v_k^S - v_k^T\right\|_{\Sigma_k^{-1}}^2. \tag{12}$$

For any positive-definite shared covariance, the unique pointwise minimizer is $v_k^S = v_k^T$, equivalently $\mu_k^S = \mu_k^T$.The covariance affects the weighting of the regression objective but not its pointwise optimum.

From stochastic transitions to deterministic matching. For two deterministic transitions with distinct endpoints, the corresponding Dirac measures are mutually singular, and their KL divergence is therefore infinite (Gray, 2011). We thus do not obtain the deterministic objective by directly taking a zero-variance KL limit. Instead, we retain the pointwise optimizer of the shared-covariance Gaussian objective and realize it through deterministic $L2$-matching.

$$\mathcal{L}_\mu = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \left\|\mu_k^S - \text{sg}[\mu_k^T]\right\|_2^2\right] = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} (\Delta t_k)^2 \left\|v_k^S - \text{sg}[v_k^T]\right\|_2^2\right]. \tag{13}$$

The rollout states and teacher predictions are detached during each optimization step. This deterministic objective preserves the pointwise teacher-matching target of the Gaussian formulation while avoiding the introduction of transition noise during training.

Alternative deterministic parameterizations. Besides matching Euler transition targets, we consider direct velocity matching and clean-latent matching. Under the linear flow interpolation

$$x_t = (1-t)x_0 + t\epsilon, \qquad v = \epsilon - x_0, \tag{14}$$

the predicted clean latent is

$$\hat{x}_{0,k}^b = x_{t_k} - t_k v_k^b, \qquad b \in \{S, T\}. \tag{15}$$

Consequently,

$$\begin{aligned} \left\|\mu_k^S - \mathrm{sg}[\mu_k^T]\right\|_2^2 &= (\Delta t_k)^2 \left\|v_k^S - \mathrm{sg}[v_k^T]\right\|_2^2, \\ \left\|\hat{x}_{0,k}^S - \mathrm{sg}[\hat{x}_{0,k}^T]\right\|_2^2 &= t_k^2 \left\|v_k^S - \mathrm{sg}[v_k^T]\right\|_2^2. \end{aligned} \tag{16}$$

The three deterministic objectives can therefore be written as

$$\mathcal{L} = \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \omega(t_k) \left\|v_k^S - \mathrm{sg}[v_k^T]\right\|_2^2\right], \tag{17}$$

with

$$\omega_v(t_k) = 1, \qquad \omega_\mu(t_k) = (\Delta t_k)^2, \qquad \omega_{x_0}(t_k) = t_k^2. \tag{18}$$

All three objectives share the pointwise optimum $v_k^S = v_k^T$, but they assign different weights to denoising timesteps and therefore need not produce identical optimization dynamics.

B Experimental Details

B.1 Training Data

For GenEval (Ghosh et al., 2023), we adopt the data splits released by Flow-GRPO (Liu et al., 2026c), consisting of 50K training prompts and 2,212 evaluation instances. For GenEval2 (Kamath et al., 2025), we use the officially released 20K synthetic prompts for training and the official set of 800 prompts for evaluation. Unless otherwise specified, all trainable methods use *MixDataset*, formed by combining the GenEval and GenEval2 training sets. Evaluation prompts are never used for training.

We construct the prompt-pair dataset $\mathcal{D} = \{(p, p^+)\}$ offline by applying each PE to the raw training prompts using the system prompt in Appendix F.2. Since `GPT-5.6 Sol` is a general-purpose model rather than a PE specifically trained for prompt rewriting, its outputs may occasionally alter or omit subject identities or attributes. We therefore apply an automatic consistency check using the system prompt in Appendix F.1. Here, we use `GPT-5.6 Sol` as judge model. Rejected prompts are regenerated until they pass this check. This preprocessing is performed once before training, so the PE need not be loaded during optimization.

B.2 Training Configurations.

All experiments are conducted on one to four nodes with 8 NVIDIA A100 GPUs. We use a unified training recipe across model families and trainable baselines. Within each backbone, all methods use the same initialization, training data, and hyper-parameters.

To improve training efficiency, we collect student rollouts using fewer denoising steps than at inference, following the denoising-reduction practice established in prior work (Liu et al., 2026c; Ping et al., 2026). Specifically, we use 4 training rollout steps for Z-Image-Turbo and FLUX.2-klein, 10 for SD3.5-M, and 14 for Z-Image, FLUX.2-klein-base, and QwenImage-2512. Here, *on-policy* refers to the provenance of the training states: they are generated by the current student under raw prompts. At every visited training state, the student and teacher are evaluated at the same latent and timestep. This does not require the training rollout and inference process to use identical timestep discretizations.

For evaluation, we set the number of inference steps to 4 for FLUX.2-klein, 8 for Z-Image-Turbo, 40 for SD3.5M, and 50 for Z-Image, FLUX.2-klein-base and QwenImage-2512 following the official settings. For evaluation CFG settings, we set 1.0 (disabled) for Z-Image-Turbo and FLUX.2-klein, 4.0 for Z-Image, FLUX.2-klein-base and QwenImage-2512, and 4.5 for SD3.5M.The other hyper-parameter settings are reported in Table 9.

The ablation studies for different loss, dataset, and training step are conducted on Z-Image-Turbo using `GPT-5.6 Sol` as PE. The CFG ablation is conducted on Z-Image with the same PE. Unless explicitly varied, all remaining settings follow the default configuration.

Table 9: **Default training hyperparameters for PE-OPSD.** All models follow the same recipe unless otherwise specified.

Configuration	Setting
Optimizer	AdamW
Learning rate	1×10^{-4}, constant without warm-up
Optimizer momentum	$(\beta_1, \beta_2) = (0.9, 0.999)$
Adam ϵ / weight decay	10^{-8} / 0
Gradient clipping	1.0
Global batch size	64
Optimization steps	1,000
LoRA rank / scaling factor	64 / 128
EMA decay γ	0.999
Training resolution	All 512×512
Training CFG	Disabled
Evaluation resolution	All 512×512
Maximum text sequence length	512
Distributed optimization	DeepSpeed ZeRO-2

Table 10: **Models and checkpoints used in our experiments.**

Model	Checkpoint
SD3.5-Medium	`stabilityai/stable-diffusion-3.5-medium`
SD3.5-Large	`stabilityai/stable-diffusion-3.5-large`
Z-Image	`Tongyi-MAI/Z-Image`
Z-Image-Turbo	`Tongyi-MAI/Z-Image-Turbo`
FLUX.1-dev	`black-forest-labs/FLUX.1-dev`
FLUX.2-klein-base-9B	`black-forest-labs/FLUX.2-klein-base-9B`
FLUX.2-klein-9B	`black-forest-labs/FLUX.2-klein-9B`
QwenImage-2512	`Qwen/Qwen-Image-2512`
PromptEnhancer-7B	`tencent/HunyuanImage-2.1/reprompt`
PromptEnhancer-32B	`PromptEnhancer/PromptEnhancer-32B`

B.3 Models

Table 10 lists the checkpoints used in our experiments.

B.4 Baselines

Inference-time PE. *Base* generates directly from the raw prompt p, whereas *Base+PE* applies the PE and conditions the frozen generator on p^+ at inference time.

Supervised fine-tuning. For SFT, we first use each frozen base model to generate pseudo-target images from enhanced prompts under its corresponding inference configuration. We then fine-tune the model with the standard flow-matching objective on raw-prompt/image pairs (p, x^+). Thus, SFT transfers enhanced-prompt behavior through offline-generated images rather than online vector-field supervision.

Off-policy distillation. Off-policy distillation uses the same configuration as PE-OPSD. The only difference is the rollout distribution: off-policy states are generated by the teacher conditioned on p^+, whereas PE-OPSD evaluates the teacher on states visited by the student conditioned on p.

B.5 BENCHMARKS

In-domain evaluation. GenEval (Ghosh et al., 2023) evaluates object presence, counting, colors, spatial relations, and attribute binding using its detector-based protocol. GenEval2 (Kamath et al., 2025) contains 800 prompts with broader visual concepts and higher compositional complexity. We follow its official Soft-TIFA evaluation, reporting the geometric mean (GE2_{GM}) for prompt-level correctness and the arithmetic mean (GE2_{AM}) for atom-level correctness. On both benchmarks, we additionally report CLIP score Hessel et al. (2021), PickScore (Kirstain et al., 2023) normalized by 26, and aesthetics score (Schuhmann, 2022).

Out-of-domain evaluation. DPG-Bench (Hu et al., 2024) evaluates dense prompt following over Global, Entity, Attribute, Relation, and Other categories. T2I-CompBench++ (Huang et al., 2025) evaluates attribute binding, numeracy, spatial and non-spatial relations, and complex compositions. For EvalMuse (Han et al., 2026), we follow the original protocol using its 200 representative prompts and the FGA-BLIP2 overall alignment score. All Overall scores are computed using the corresponding official evaluation procedures.

C HUMAN EVALUATION PROTOCOL

Evaluation data and comparisons. We conduct human evaluation on all 800 prompts from the GenEval2 test set using Z-Image-Turbo as the generator and `GPT-5.6 Sol` as the PE. For each prompt, we compare Base against Base+PE, SFT, off-policy distillation, and PE-OPSD.

Evaluation interface and aggregation. Each comparison presents two generated images side by side, with the assignment of systems to the left and right positions randomized independently for every pair. Model identities and method names are hidden from the assessors, while the corresponding raw prompt is displayed above the images. Assessors provide two independent judgments (Left better or Right better): (1) *Prompt Fidelity*, indicating which image more faithfully satisfies the objects, attributes, counts, and relationships specified by the raw prompt; and (2) *Visual Appeal*, indicating which image has better overall perceptual quality and aesthetics. For each criterion, the assessor selects either the left or right image. For the result aggregation, each image pair is independently evaluated by all three assessors for both criteria. We determine the preference for each pair by majority vote.

Assessors and quality control. We recruit three professional assessors who are formally contracted and compensated at locally competitive rates. Before participating in the study, all assessors are informed of the task and potential exposure to generated content. They receive detailed criterion-specific instructions and complete a qualification test covering representative evaluation cases.

D ADDITIONAL EXPERIMENTAL RESULTS

D.1 TRAINING EFFICIENCY

Table 11 reports the method-specific cost of post-training. Although SFT has a substantially lower per-step cost, it requires generating pseudo-target images for the entire training set. This preprocessing dominates its total cost for SD3.5-M and Z-Image, for which PE-OPSD reduces the training time from 7.0 to 3.1 hours and from 28.4 to 13.6 hours, respectively. For Z-Image-Turbo, few-step sampling makes pseudo-target generation inexpensive, and SFT is slightly faster overall. However, SFT yields lower performance than both off-policy distillation and PE-OPSD (see Table 2).

PE-OPSD and off-policy distillation have identical training costs because they use the same number of rollout and teacher evaluations, differing only in whether states are collected from the student or teacher trajectory. Overall, PE-OPSD avoids the additional image-generation and storage requirements of SFT while incurring the expected cost of online rollout supervision. Offline enhanced-prompt construction is shared by all trainable methods and is therefore excluded from this method-specific comparison.

Table 11: **Training efficiency in our experiments.** One-step denotes the wall-clock time per optimization step under the default configuration, and Full denotes the cost of 1,000 optimization steps. For SFT, Full† is reported as training time + offline pseudo-target generation time.

Model	SFT		Off-policy		PE-OPSD	
	One-step	Full†	One-step	Full	One-step	Full
SD3.5-M	$1.47s$	$0.4{+}6.6h$	$11.02s$	$3.1h$	$11.02s$	$3.1h$
Z-Image	$3.14s$	$0.9{+}27.5h$	$49.15s$	$13.6h$	$49.15s$	$13.6h$
Z-Image-Turbo	$3.14s$	$0.9{+}2.5h$	$14.31s$	$4.0h$	$14.31s$	$4.0h$

D.2 Inference Latency

Table 12 compares deployment latency with and without inference-time PE. Because PE-OPSD generates directly from raw prompts using the original inference pipeline, it retains the latency of Base across all evaluated backbones. In contrast, inference-time PE introduces a substantial fixed overhead, particularly for efficient few-step generators. Relative to `PromptEnhancer-7B`, PE-OPSD provides $1.52\times$–$12.21\times$ speedups; with `PromptEnhancer-32B`, the speedups increase to $3.31\times$–$50.67\times$. The largest gains occur on Z-Image-Turbo and FLUX.2-klein, where prompt rewriting is considerably more expensive than image generation itself. These results demonstrate that PE-OPSD preserves the benefits of enhanced prompts without adding per-request deployment latency.

Table 12: **Inference latency and speedup across different settings.** Latency is the average generation time per image in seconds. +PE includes both prompt rewriting and image generation, whereas +Ours generates directly from the raw prompt without invoking PE.

Model	Latency (s)			Speedup
	Base	+PE	+Ours	
`PromptEnhancer-7B as PE`				
SD3.5-M	2.70	10.36	2.70	3.84×
Z-Image	9.69	19.00	9.69	1.96×
Z-Image-Turbo	0.88	8.53	0.88	9.69×
FLUX.2-klein	0.67	8.18	0.67	12.21×
FLUX.2-klein-base	14.45	21.97	14.45	1.52×
QwenImage-2512	13.38	22.09	13.38	1.65×
`PromptEnhancer-32B as PE`				
SD3.5-M	2.70	35.91	2.70	13.30×
Z-Image	9.69	43.74	9.69	4.51×
Z-Image-Turbo	0.88	34.21	0.88	38.88×
FLUX.2-klein	0.67	33.95	0.67	50.67×
FLUX.2-klein-base	14.45	47.77	14.45	3.31×
QwenImage-2512	13.38	47.33	13.38	3.54×

D.3 Effect of EMA Teacher

As shown in Figure 8 and Figure 9, we study the effect of EMA teacher using four settings: a frozen Base teacher without EMA updates (No EMA), $\gamma = 0.99$, $\gamma = 0.9$, and our default $\gamma = 0.999$. We conduct the experiments using Z-Image-Turbo with 500 training steps.

The frozen teacher performs similarly to the default setting during early training but reaches a lower performance ceiling, indicating that allowing the teacher to evolve with the student provides stronger supervision at later stages. With $\gamma = 0.99$, generated images remain visually coherent, although performance is slightly lower than with $\gamma = 0.999$.

A more aggressive update with $\gamma = 0.9$ substantially degrades generation quality. Although its training loss decreases rapidly, generated images contain pronounced artifacts and evaluation scores fall below Base. This suggests that an overly responsive teacher becomes too tightly coupled to the student and provides insufficiently stable targets. We therefore use $\gamma = 0.999$, which balances teacher adaptation with temporal stability.

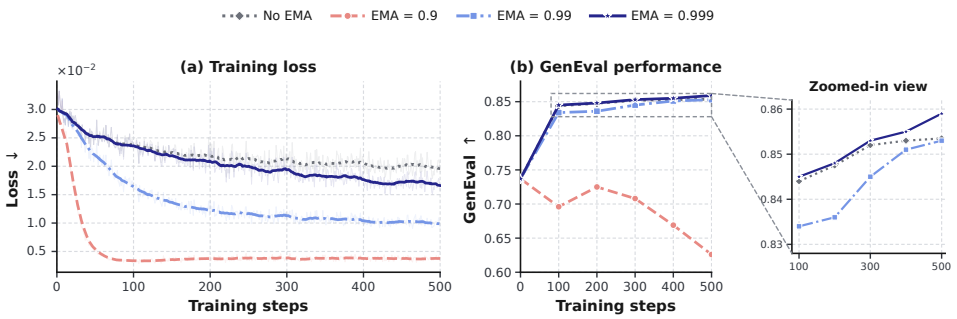

Figure 8: **Training dynamics and GenEval performance with different EMA settings.** Left: the loss curves where faint lines denote raw losses and bold lines show a the moving average; Right: the corresponding GenEval performance curves.

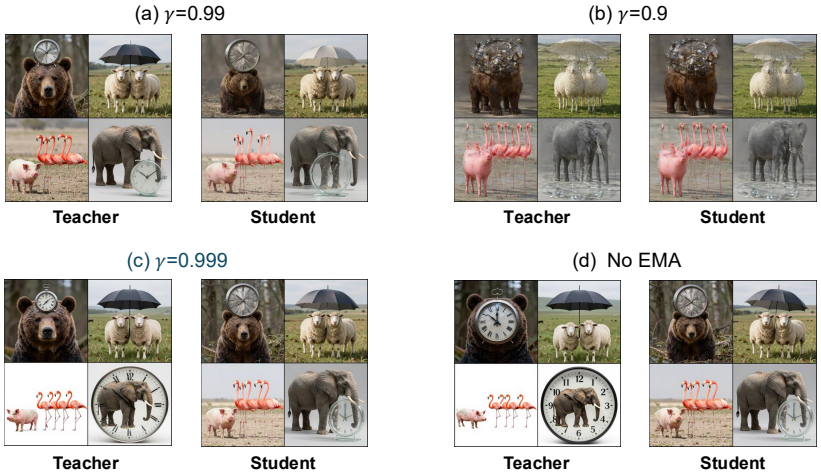

Figure 9: **Visual examples.** We visualize the samples generated by the student and teacher across different EMA settings at 500 training steps.

D.4 Detailed Results of Scalability

Table 13 reports the complete results on larger models. PE-OPSD achieves the highest aggregate prompt-fidelity improvement across all three backbones and consistently leads on GE and $GE2_{GM}$. It also improves aggregate visual appeal over Base, although Base+PE remains marginally better on the two FLUX variants. On QwenImage-2512, PE-OPSD obtains the best aggregate results for both objectives, further supporting its applicability to larger models.

Table 13: **Results on scaling up to large models.** We use `GPT-5.6 Sol` as PE. PickScore is normalized by 26; Aes. denotes aesthetics; **Bold:** best; Underlined: second-best.

Method	GenEval (GE) Task				GenEval2 (GE2) Task					$\mathbb{I}_{PF}$	$\mathbb{I}_{VA}$
	GE	PickScore	CLIP	Aes.	$GE2_{GM}$	$GE2_{AM}$	PickScore	CLIP	Aes.		
FLUX.2-klein-base (9B)	0.775	0.892	0.303	5.140	0.359	0.773	0.880	0.328	5.310	0.00%	0.00%
+PE	0.862	0.910	0.302	**5.467**	0.442	**0.838**	0.901	0.330	**5.658**	8.61%	**4.33%**
+Ours	**0.873**	**0.913**	**0.305**	5.421	**0.444**	0.831	**0.905**	**0.333**	5.628	**9.20%**	4.16%
Δ (vs Base)	+0.098	+0.021	+0.002	+0.281	+0.085	+0.058	+0.025	+0.005	+0.318	+9.20%	+4.16%
FLUX.2-klein (9B)	0.856	0.911	0.299	5.288	0.348	0.797	0.901	**0.329**	5.430	0.00%	0.00%
+PE	0.859	**0.915**	0.300	**5.474**	0.381	**0.827**	0.905	0.327	5.682	2.66%	**2.26%**
+Ours	**0.867**	0.912	**0.301**	5.433	**0.434**	0.825	**0.907**	0.328	**5.700**	**5.98%**	2.12%
Δ (vs Base)	+0.011	+0.001	+0.002	+0.145	+0.086	+0.028	+0.006	-0.001	+0.270	+5.98%	+2.12%
QwenImage-2512 (20B)	0.620	0.895	0.283	5.377	0.163	0.671	0.894	0.303	5.822	0.00%	0.00%
+PE	0.839	0.916	0.297	**5.462**	0.380	0.813	0.908	0.321	5.782	40.1%	1.20%
+Ours	**0.857**	**0.920**	**0.302**	5.421	**0.399**	**0.820**	**0.916**	**0.324**	**5.857**	**43.8%**	**1.67%**
Δ (vs Base)	+0.237	+0.025	+0.019	+0.044	+0.236	+0.149	+0.022	+0.021	+0.035	+43.8%	+1.67%

D.5 More Analysis in Table 4

Robustness across PEs. Table 4 shows that the performance of SFT, inference-time PE, and off-policy distillation varies with the chosen PE, whereas PE-OPSD consistently provides the strongest aggregate results. In particular, PE-OPSD improves $\mathbb{I}_{\mathrm{PF}}$ over off-policy distillation by 1.80–3.78 percentage points and $\mathbb{I}_{\mathrm{VA}}$ by 0.18–0.33 points across the three PEs. Since the two distillation methods share the teacher condition, objective, and optimization budget, this consistent margin supports the importance of supervising the student on states visited by its own raw-prompt trajectories.

Effect of the teacher condition. PE-OPSD also exceeds inference-time PE in both aggregate indices while requiring only raw prompts at deployment. Notably, its largest fidelity margin over Base+PE occurs with `PromptEnhancer-7B`, whose direct inference-time improvement is the weakest among the tested PEs. This indicates that the transferred benefit is not determined solely by the PE's one-shot inference performance. Meanwhile, `PromptEnhancer-32B` produces the strongest fidelity supervision, whereas `GPT-5.6 Sol` yields the largest visual-appeal improvement. Overall, PE-OPSD remains effective across PEs with substantially different capacities and enhancement behaviors.

D.6 Different PEs

Additional experimental results for different PEs. We further visualize the training dynamics with three PEs in our experiments across SD3.5-M, Z-Image, and Z-Image-Turbo. As shown in Figure 10, the PE-OPSD loss consistently decreases and stabilizes within 1,000 training steps for all PEs and models. These similar optimization trends demonstrate that PE-OPSD remains stable across different PEs and does not rely on a specific PE.

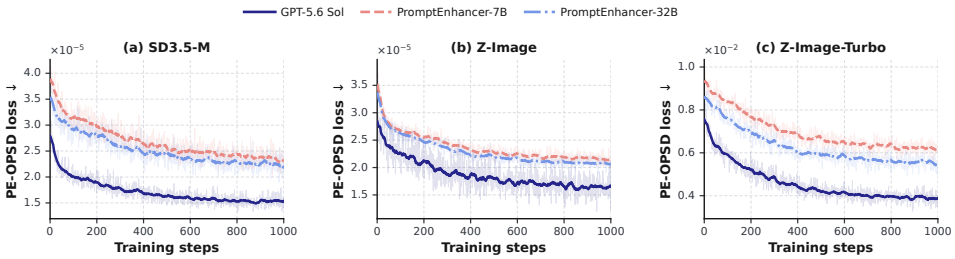

Figure 10: **Training dynamics with different PEs.** We report the PE-OPSD loss over 1,000 training steps. Faint lines denote raw losses, while bold lines show a the moving average.

Prompt examples. Figures 11–13 show examples of raw prompts and their enhanced versions produced by different PEs.

Raw Prompt

five bears and a donut.

Enhanced Prompt

Five bears gather in a loose semicircle around a single oversized glazed donut resting on the ground. Each bear has thick, textured brown fur, rounded ears, broad paws, and a distinct curious expression, with two bears seated at the front, two standing behind them, and the fifth leaning forward to inspect the pastry. The donut features a golden-brown surface, glossy pink icing, and scattered rainbow sprinkles, contrasting vividly with the bears' earthy coloring. Soft forest vegetation, moss-covered stones, and tall trees fill the background, while warm filtered sunlight creates gentle shadows and highlights the fur and shiny glaze. Whimsical, highly detailed storybook illustration with naturalistic anatomy and cinematic lighting.

Figure 11: **Comparison between the raw prompt and the enhanced prompt.** Here, we use `GPT-5.6 Sol` as PE.

Raw Prompt

five bears and a donut.

Enhanced Prompt

A group of five cartoon bears and one donut are presented in a cheerful arrangement against a plain, light-colored background. The bears are positioned in a line, with three standing and two sitting. The bear on the far left is a tall, slender figure with light brown fur, standing on its hind legs and waving a paw. Next to it, another tall bear with chocolate-brown fur stands with its head tilted curiously. In the center, a large, round brown bear sits comfortably, looking directly forward with a friendly expression. To its right, a shorter, honey-colored bear sits alongside the donut, looking up at it. On the far right, the fifth bear, a small grizzly-style bear with darker fur, sits with its paws on its knees. In front of the central sitting bear rests a large, classic ring donut, coated in a vibrant pink frosting and decorated with a scattering of multi-colored sprinkles. The image is rendered in a clean, digital illustration style with bold outlines and soft shading.

Figure 12: **Comparison between the raw prompt and the enhanced prompt.** Here, we use `PromptEnhancer-7B` as PE.

Raw Prompt

five bears and a donut.

Enhanced Prompt

A group of five small, plush bears is arranged in a circle on a plain, light-colored surface, with a single large donut placed among them. The donut, positioned in the center, is perfectly round with a golden-brown, fried surface and is covered in a thick, rainbow-colored glaze topped with colorful sprinkles. Surrounding the donut are the five bears, each with a distinct color and texture; one brown bear holds the donut, while a black bear, a beige bear, and two other brown bears look towards it. The bears are made of a fluffy, plush material, featuring simple stitched noses and button eyes. This image presents a photography style with a shallow depth of field, emphasizing the subjects.

Figure 13: **Comparison between the raw prompt and the enhanced prompt.** Here, we use `PromptEnhancer-32B` as PE.

E Additional Benchmark Results

E.1 GenEval Details

In Table 14 and Table 15, we provide the detailed GenEval performance breakdown for different models and PEs. In particular, we report the fine-grained performance in Single Object, Two Object, Counting, Colors, Position, and Attribute Binding. We also report the Overall score.

E.2 GenEval2 Details

In Table 16 and Table 17, we provide the detailed GenEval2 performance breakdown for different models and PEs. In particular, we report the fine-grained performance in Object, Attribute, Count, Position, and Verb. We also report the overall soft-tifa score in $\text{GenEval2}_{\text{AM}}$ and $\text{GenEval2}_{\text{GM}}$.

Table 14: **Detailed GenEval performance breakdown.** We report the fine-grained performance in Single Object, Two Object, Counting, Colors, Position, and Attribute Binding. We also report the Overall score. Here, we use `GPT-5.6 Sol` as PE.

Method	Single Obj	Two Obj	Counting	Color	Position	Attr Binding	Overall
SD3.5-M (2.5B)							
Base	0.975	0.778	0.613	0.787	0.223	0.472	0.628
Base+PE	0.959	0.838	0.634	0.838	0.603	0.635	0.743
SFT	0.972	0.856	0.691	0.832	0.512	0.500	0.718
Off-Policy Distillation	0.981	0.846	0.628	0.838	0.698	0.650	0.770
PE-OPSD (Ours)	0.988	0.886	0.631	0.878	0.698	0.710	0.797
Z-Image (6B)							
Base	0.959	0.801	0.569	0.809	0.338	0.480	0.650
Base+PE	0.978	0.879	0.619	0.888	0.755	0.743	0.810
SFT	0.972	0.884	0.697	0.902	0.780	0.698	0.821
Off-Policy Distillation	0.988	0.886	0.641	0.899	0.762	0.767	0.824
PE-OPSD (Ours)	0.988	0.884	0.656	0.918	0.823	0.805	0.846
Z-Image-Turbo (6B)							
Base	0.988	0.833	0.759	0.859	0.460	0.588	0.737
Base+PE	0.975	0.864	0.803	0.923	0.770	0.787	0.850
SFT	0.981	0.732	0.625	0.894	0.708	0.677	0.766
Off-Policy Distillation	0.984	0.866	0.778	0.907	0.805	0.777	0.851
PE-OPSD (Ours)	0.969	0.879	0.794	0.915	0.797	0.835	0.863
FLUX.2-klein-base (9B)							
Base	0.994	0.851	0.706	0.904	0.632	0.603	0.775
Base+PE	0.984	0.904	0.797	0.928	0.820	0.757	0.862
PE-OPSD (Ours)	0.991	0.904	0.819	0.912	0.840	0.785	0.873
FLUX.2-klein (9B)							
Base	0.994	0.912	0.847	0.894	0.733	0.787	0.856
Base+PE	0.991	0.896	0.775	0.912	0.835	0.757	0.859
PE-OPSD (Ours)	0.988	0.874	0.747	0.931	0.845	0.823	0.867
QwenImage-2512 (20B)							
Base	0.984	0.798	0.331	0.832	0.302	0.500	0.620
Base+PE	0.975	0.904	0.694	0.891	0.755	0.818	0.839
PE-OPSD (Ours)	0.981	0.899	0.706	0.910	0.810	0.833	0.857

Table 15: **Detailed GenEval performance breakdown on Z-image with other PEs.** We report the fine-grained performance in Single Object, Two Object, Counting, Colors, Position, and Attribute Binding. We also report the Overall score.

Method	Single Obj	Two Obj	Counting	Color	Position	Attr Binding	Overall
PromptEnhancer-7B as PE							
Base	0.959	0.801	0.569	0.809	0.338	0.480	0.650
Base+PE	0.947	0.801	0.588	0.750	0.505	0.555	0.684
SFT	0.966	0.871	0.762	0.835	0.627	0.560	0.763
Off-Policy Distillation	0.978	0.856	0.716	0.856	0.613	0.623	0.767
PE-OPSD (Ours)	0.978	0.864	0.647	0.872	0.610	0.740	0.782
PromptEnhancer-32B as PE							
Base	0.959	0.801	0.569	0.809	0.338	0.480	0.650
Base+PE	0.984	0.854	0.675	0.902	0.797	0.690	0.815
SFT	0.972	0.884	0.741	0.888	0.825	0.680	0.829
Off-Policy Distillation	0.981	0.889	0.725	0.896	0.853	0.718	0.842
PE-OPSD (Ours)	0.991	0.904	0.666	0.941	0.882	0.810	0.868

Table 16: **Detailed GenEval2 performance breakdown.** We report the fine-grained performance in Object, Attribute, Count, Position, and Verb. We also report the overall soft-tifa score in $\text{GenEval2}_{\text{AM}}$ and $\text{GenEval2}_{\text{GM}}$. Here, we use `GPT-5.6 Sol` as PE.

Method	Object	Attribute	Count	Position	Verb	$\text{GenEval2}_{\text{AM}}$	$\text{GenEval2}_{\text{GM}}$
SD3.5-M (2.5B)							
Base	0.859	0.672	0.431	0.397	0.161	0.633	0.176
Base+PE	0.868	0.751	0.476	0.521	0.207	0.680	0.220
SFT	0.835	0.703	0.478	0.438	0.124	0.656	0.207
Off-Policy Distillation	0.850	0.715	0.461	0.483	0.195	0.671	0.223
PE-OPSD (Ours)	0.868	0.758	0.477	0.489	0.179	0.682	0.226
Z-Image (6B)							
Base	0.927	0.844	0.599	0.607	0.294	0.761	0.306
Base+PE	0.973	0.923	0.637	0.826	0.459	0.827	0.404
SFT	0.981	0.906	0.643	0.816	0.323	0.834	0.403
Off-Policy Distillation	0.975	0.931	0.625	0.814	0.320	0.828	0.395
PE-OPSD (Ours)	0.975	0.938	0.634	0.816	0.381	0.831	0.406
Z-Image-Turbo (6B)							
Base	0.970	0.763	0.665	0.624	0.196	0.783	0.341
Base+PE	0.973	0.887	0.688	0.845	0.285	0.843	0.479
SFT	0.929	0.866	0.601	0.799	0.205	0.788	0.413
Off-Policy Distillation	0.968	0.892	0.640	0.821	0.297	0.821	0.437
PE-OPSD (Ours)	0.975	0.894	0.686	0.851	0.353	0.844	0.501
FLUX.2-klein-base (9B)							
Base	0.929	0.863	0.567	0.710	0.340	0.773	0.359
Base+PE	0.962	0.950	0.647	0.838	0.477	0.838	0.442
PE-OPSD (Ours)	0.956	0.945	0.641	0.830	0.399	0.831	0.444
FLUX.2-klein (9B)							
Base	0.954	0.889	0.593	0.742	0.315	0.797	0.348
Base+PE	0.961	0.952	0.609	0.843	0.421	0.827	0.381
PE-OPSD (Ours)	0.960	0.947	0.616	0.846	0.362	0.825	0.434
QwenImage-2512 (20B)							
Base	0.930	0.593	0.526	0.510	0.270	0.671	0.163
Base+PE	0.977	0.848	0.620	0.843	0.423	0.813	0.380
PE-OPSD (Ours)	0.978	0.882	0.625	0.851	0.397	0.820	0.399

Table 17: **Detailed GenEval2 performance breakdown on Z-Image with other PEs.** We report the fine-grained performance in Object, Attribute, Count, Position, and Verb. We also report the overall soft-tifa score in $\text{GenEval2}_{\text{AM}}$ and $\text{GenEval2}_{\text{GM}}$.

Method	Object	Attribute	Count	Position	Verb	$\text{GenEval2}_{\text{AM}}$	$\text{GenEval2}_{\text{GM}}$
PromptEnhancer-7B as PE							
Base	0.927	0.844	0.599	0.607	0.294	0.761	0.306
Base+PE	0.962	0.818	0.653	0.723	0.471	0.801	0.375
SFT	0.975	0.865	0.660	0.765	0.340	0.819	0.400
Off-Policy Distillation	0.975	0.832	0.649	0.720	0.274	0.804	0.380
PE-OPSD (Ours)	0.979	0.875	0.636	0.784	0.388	0.819	0.422
PromptEnhancer-32B as PE							
Base	0.927	0.844	0.599	0.607	0.294	0.761	0.306
Base+PE	0.966	0.924	0.664	0.864	0.543	0.844	0.470
SFT	0.981	0.925	0.672	0.848	0.339	0.845	0.440
Off-Policy Distillation	0.970	0.940	0.662	0.832	0.382	0.843	0.459
PE-OPSD (Ours)	0.982	0.952	0.655	0.845	0.430	0.846	0.476

E.3 EvalMuse Details

In Table 18, we provide the detailed performance breakdown for different models. In particular, we report the fine-grained performance in Attribute, Location, Color, Object, Material, A./H. (Animal/Human), Food, Shape, Activity, Spatial, and Counting. We also report the overall score following the official guidance (Han et al., 2026).

Table 18: **Detailed EvalMuse performance breakdown.** We report the fine-grained performance in Attribute, Location, Color, Object, Material, A./H. (Animal/Human), Food, Shape, Activity, Spatial, and Counting. We also report the overall score. **Bold:** best in Overall.

Method	Attribute	Location	Color	Object	Material	A./H.	Food	Shape	Activity	Spatial	Counting	Overall
SD3.5-M	0.801	0.724	0.570	0.675	0.542	0.524	0.654	0.826	0.539	0.459	0.234	3.199
+Ours	0.820	0.763	0.634	0.710	0.587	0.562	0.697	0.832	0.551	0.588	0.283	**3.368**
Z-Image	0.797	0.723	0.618	0.709	0.620	0.608	0.656	0.806	0.582	0.655	0.314	3.340
+Ours	0.809	0.768	0.694	0.738	0.681	0.654	0.716	0.811	0.596	0.643	0.384	**3.561**
Z-Image-Turbo	0.800	0.762	0.711	0.727	0.707	0.620	0.699	0.798	0.589	0.641	0.340	3.515
+Ours	0.807	0.753	0.626	0.729	0.639	0.632	0.735	0.806	0.620	0.584	0.337	**3.534**
FLUX.2-klein-base	0.834	0.784	0.707	0.749	0.680	0.659	0.708	0.765	0.644	0.661	0.311	3.638
+Ours	0.829	0.797	0.728	0.755	0.691	0.700	0.733	0.798	0.664	0.680	0.367	**3.733**
FLUX.2-klein	0.832	0.778	0.716	0.761	0.686	0.691	0.728	0.834	0.661	0.675	0.337	**3.724**
+Ours	0.832	0.788	0.725	0.760	0.691	0.687	0.726	0.810	0.660	0.686	0.333	3.710
QwenImage-2512	0.817	0.757	0.616	0.720	0.662	0.685	0.708	0.835	0.618	0.605	0.280	3.485
+Ours	0.829	0.798	0.749	0.772	0.750	0.693	0.736	0.837	0.643	0.725	0.397	**3.756**

F System Prompts

F.1 System Prompt for Consistency Filtering

For GPT-generated enhanced prompts, we employ an consistency judge to identify expansions that alter or omit critical semantics from the original prompt. The judge verifies exact counts, object identities, attributes, actions, and spatial relations, and returns a binary decision in a structured JSON format. The exact system prompt is provided below.

System prompt for consistency filtering

You are a strict consistency judge for enhanced text-to-image prompts.

The original prompt is the source of truth. The candidate enhanced prompt is consistent only if it explicitly preserves every critical fact:

- exact counts and object identities;
- colors, visual material appearances, textures, and other attributes;
- actions and subject/object roles;
- spatial relations and ordering.

Do not treat a fact as preserved when it is merely implied. Synonyms and grammatical number words are acceptable when their meaning is unambiguous. Additional composition, lighting, style, and background details are acceptable only when they do not alter or contradict an original fact. Be especially strict about swapped roles, changed counts, merged object groups, conflicting attributes, and missing relations.

Do not relax colors, counts, identities, actions, or relations. In particular, spots, reflections, or small localized details do not satisfy an overall color requirement when the object is explicitly described as a different base color.

Return exactly one JSON object with no Markdown:
`{"consistent": true, "issues": []}`
or
`{"consistent": false, "issues": ["specific issue", "..."]}`

Judge only. Do not rewrite, complete, or suggest a replacement prompt.

F.2 System prompts for generating enhanced prompts.

we provide the system prompt used by `GPT-5.6 Sol` to generate enhanced prompts. Given a raw prompt, the enhancer is instructed to produce only the expanded caption without auxiliary explanations or Markdown formatting. The instruction encourages a hierarchical description while explicitly preserving the objects, attributes, spatial relations, rendered text, and intellectual-property subjects specified in the original prompt.

System prompt for GPT-5.6 Sol

You are an expert in writing prompts for image generation. I will give you a sentence, and you are to expand this sentence into a detailed caption for generating an image. And the captions must follow the rules listed below.

I. Sentence Structures The captions follow a consistent, hierarchical structure that moves from a general overview to specific details.

1. The Opening Statement: General Overview
2. The Body: Systematic and Spatially Organized Description
3. Hierarchical Object Description: From Whole to Parts
4. The Concluding Statement: Stylistic Identification

II. Grammatical Rules The grammar is precise, descriptive, and maintains an objective tone.

1. Tense: Consistent Present Tense
2. Voice: Mix of Active and Passive
3. Prepositional Phrases for Precision
4. Participial Phrases for Efficient Detail
5. Rich and Specific Adjectives
6. Precision and Hedging Language
7. Complex and Compound Sentences

Key constraints:

1. Only provide the final captions, do not use markdown format.
2. The expanded captions must follow the rules listed above.
3. The expanded captions should adhere to the original sentence, especially the subject and the subject's attributes, including color, size, spatial relationships, etc.
4. You can use your world knowledge to expand some professional terminology to proper explanations that suitable for image generation models.
5. If the style of original sentence is not mentioned, you should assume it is a photography style. And you can infer the style from the context of the sentence if the photography style is not suitable.
6. Describe the scene or subject directly, do not use "The image", "The composition", "The scene" and similar words in the beginning of the captions.
7. If the original sentence has a IP subject, you should keep the IP subject in the expanded captions, and describe the background of the IP in the expanded captions.
8. If the original sentence has a text that need to be rendered, you should keep the text in the expanded captions, and format text as "rendered text".

Next, I will give you my sentence. Please provide the expanded captions:

G Additional Qualitative Examples

We provide more visual examples across different models and methods in Figure 14–Figure 16.

H Limitations and Future Work

PE-OPSD uses enhanced prompts as privileged supervision, and its learning signal is therefore influenced by the quality and semantic faithfulness of the selected PE. Although our consistency filtering and experiments with multiple PEs demonstrate robust improvements, inaccurate or overly specific rewrites may still introduce undesirable supervision. Future work could investigate confidence-aware filtering, agreement across multiple PEs, and adaptive supervision that emphasizes reliable prompt elements.

PE-OPSD shifts computation from per-request inference to offline prompt construction and post-training. Reducing this cost through selective timestep supervision and fewer teacher evaluations is a promising direction. More broadly, extending privileged-condition distillation beyond flow-matching text-to-image models may further establish its applicability across generative paradigms.

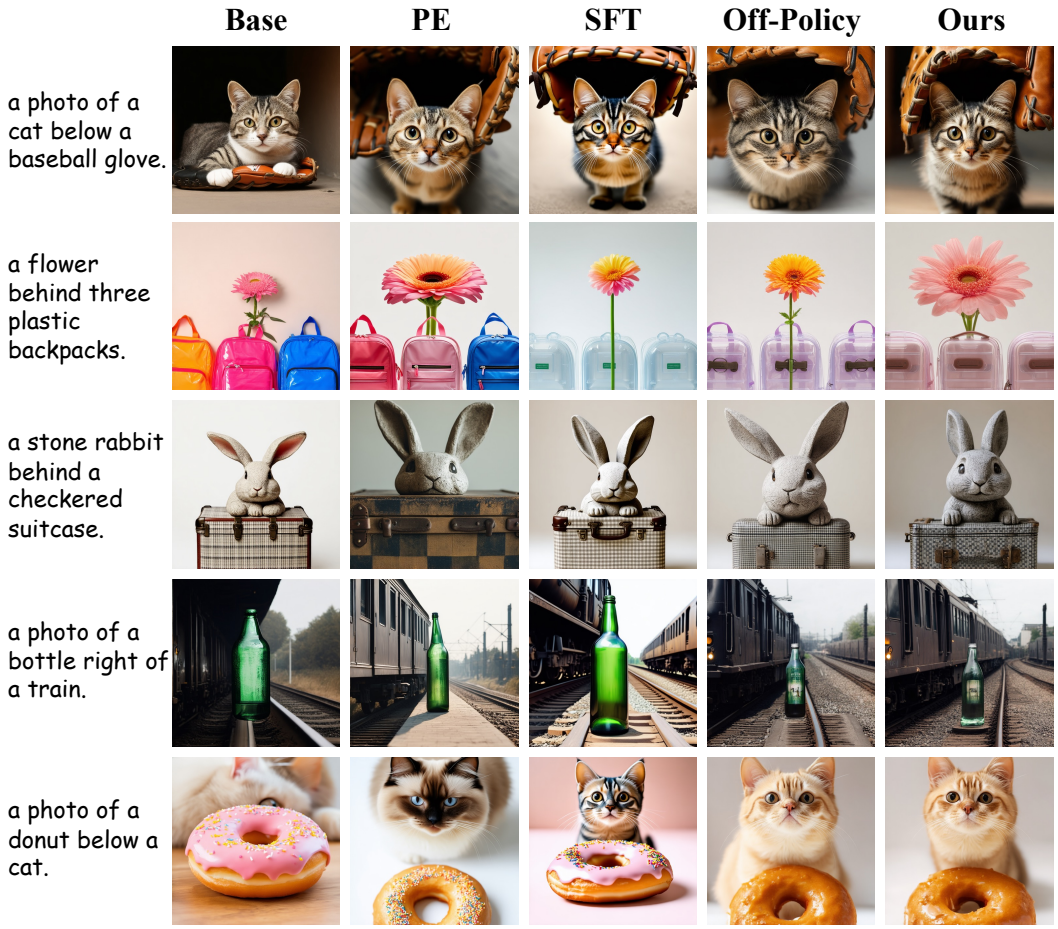

Figure 14: **Visual examples in SD3.5-M.** We use the `GPT-5.6 Sol` as PE.

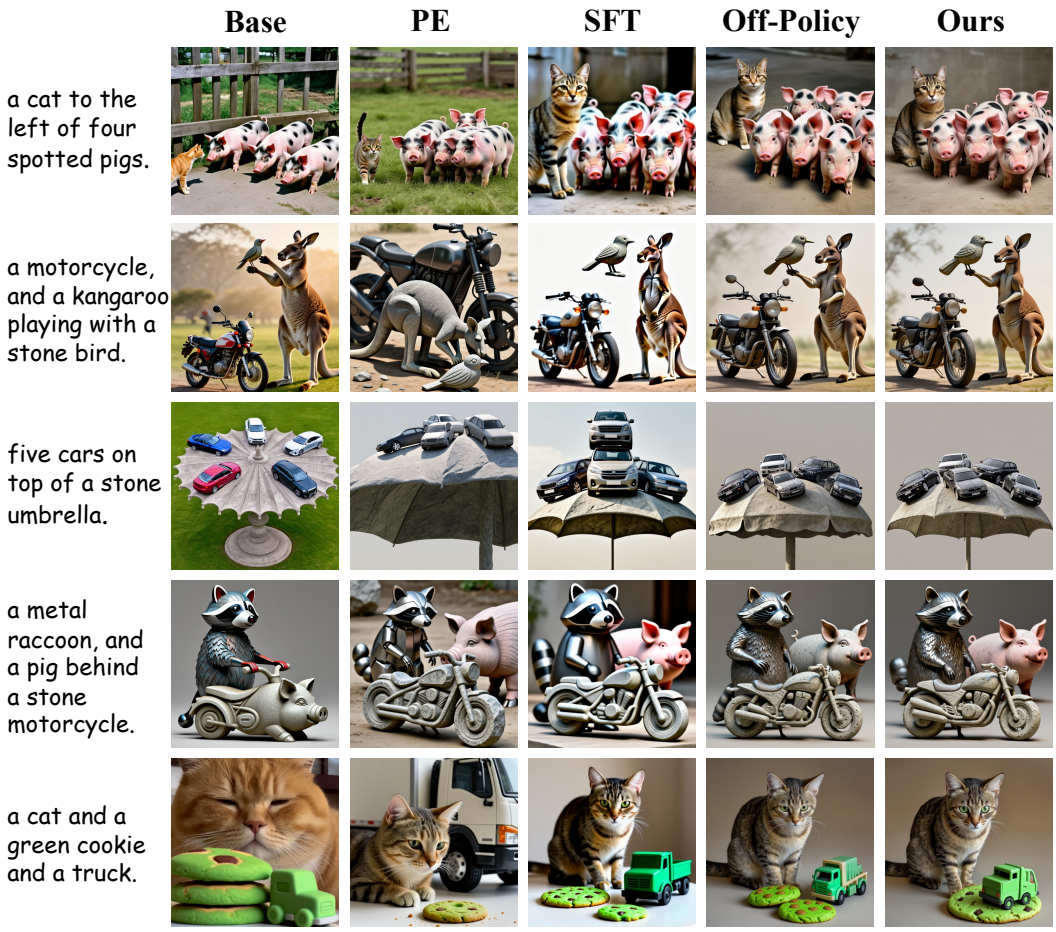

Figure 15: **Visual examples in Z-Image.** We use the `GPT-5.6 Sol` as PE.

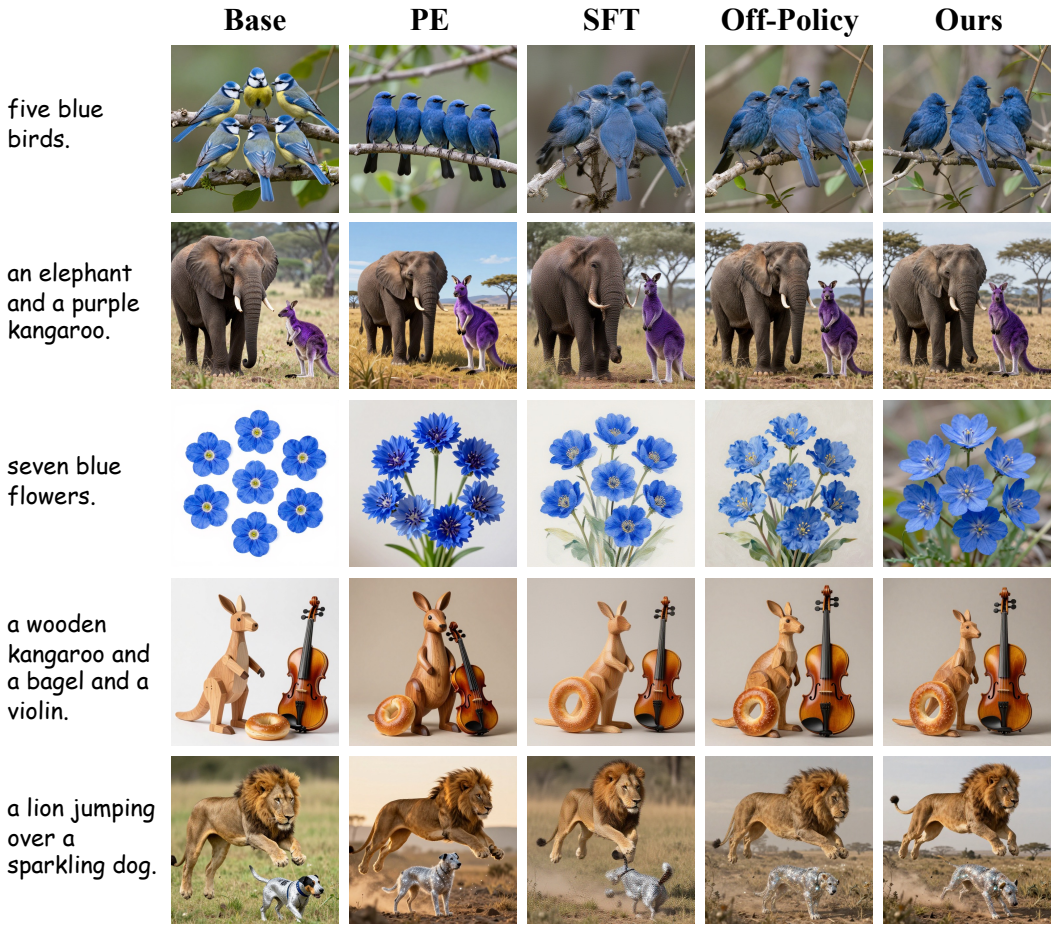

Figure 16: **Visual examples in Z-Image-Turbo.** We use the `GPT-5.6 Sol` as PE.